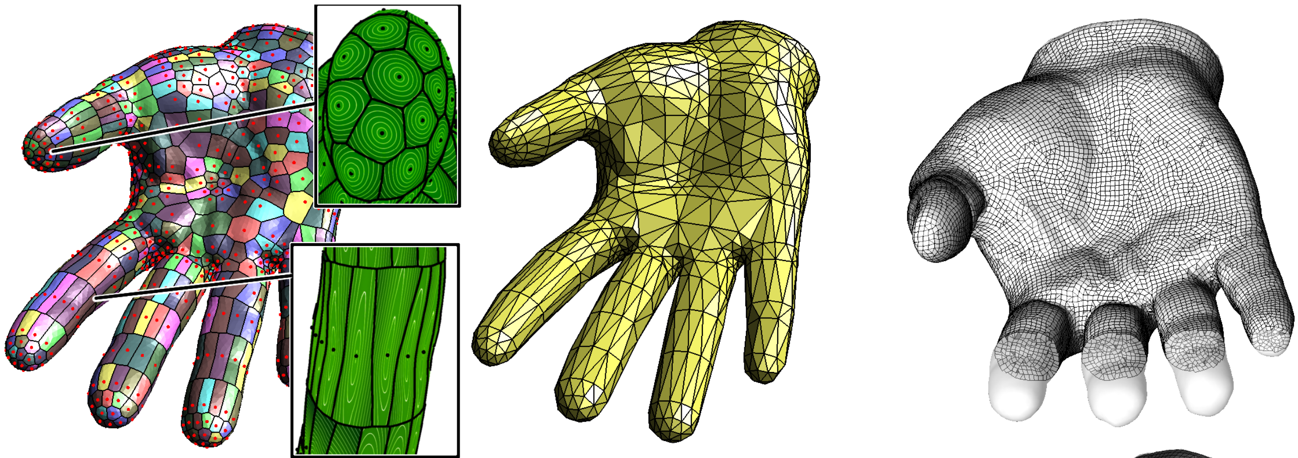
Lp Centroidal Voronoi Tesselation and its applications

Bruno Lévy, Yang Liu

L_p Centroidal Voronoi Tessellation and its Applications

Bruno Lévy* Yang Liu*



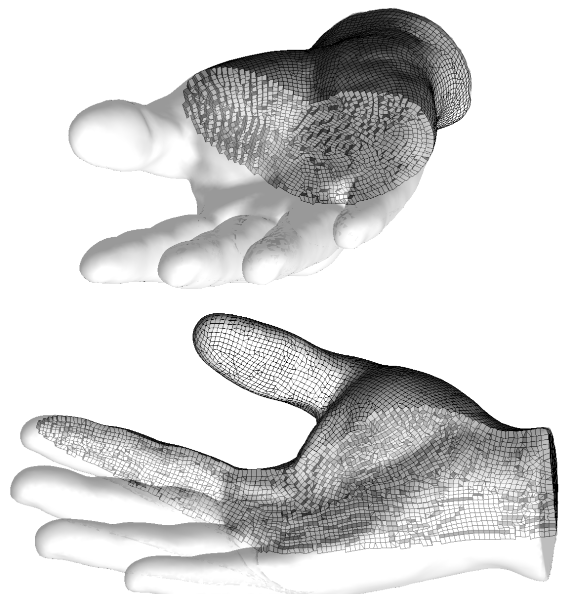


Figure 1: *Two applications of L_p-CVT. Top left: restricted L_p-CVT for anisotropic surface remeshing; right: L_p-CVT for hex-dominant meshing.*

Abstract

This paper introduces L_p-Centroidal Voronoi Tessellation (L_p-CVT), a generalization of CVT that minimizes a higher-order moment of the coordinates on the Voronoi cells. This generalization allows for aligning the axes of the Voronoi cells with a predefined background tensor field (anisotropy). L_p-CVT is computed by a quasi-Newton optimization framework, based on closed-form derivations of the objective function and its gradient. The derivations are given for both surface meshing ($\mathbf{\Omega}$ is a triangulated mesh with per-facet anisotropy) and volume meshing ($\mathbf{\Omega}$ is the interior of a closed triangulated mesh with a 3D anisotropy field). Applications to anisotropic, quad-dominant surface remeshing and to hex-dominant volume meshing are presented. Unlike previous work, L_p-CVT captures sharp features and intersections without requiring any pre-tagging.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; Algorithms

Keywords: Centroidal Voronoi Tessellation, Anisotropic Meshing, quad-dominant meshing, Hex-dominant meshing

1 Introduction

Meshing a domain consists in partitioning it into a set of cells that satisfy certain geometric requirements specified by the application.

*INRIA-ALICE e-mail: {Bruno.Levy | Yang.Liu}@inria.fr

A family of methods, called *variational*, is based on optimizing an objective function of the coordinates at the vertices. This family of methods can efficiently and robustly generate isotropic simplicial meshes [Du and Wang 2003; Alliez et al. 2005; Tournois et al. 2009]. A key ingredient of these methods is the notion of Centroidal Voronoi Tessellation (CVT) [Du et al. 1999] and Optimal Delaunay Triangulation (ODT) [Chen and Xu 2004], used to define the objective function.

This paper deals with generalizing CVT for anisotropic and hexahedral meshing. Hexahedral meshes are the preferred representations for certain finite element simulations and numerical analysis, such as CFD (computational fluid dynamics), reservoir engineering in oil exploration, or mechanics within highly elastic and plastic domains, for which they provide more reliable simulations while reducing the total number of elements. However, meshing with hexahedral elements is a challenging problem [Shepherd and Johnson 2008]. It is still performed partly manually and takes several orders of magnitude longer than meshing with tetrahedra. Designing a hex-meshing algorithm that is *robust*, *controllable* and *automatic* (see below) is still an open problem :

- **robust:** A hex-meshing algorithm needs to be independent of the commonly encountered degeneracies and singularities, including non-conforming triangles (or artificial borders), degenerate triangles, creases and self-intersections;
- **controllable:** Finite Element Modeling applications require a fine level of control over mesh generation, such as fitting a user-defined background anisotropy field, that specifies both the orientation and element sizes;
- **automatic:** The algorithm needs to be able to operate on raw data, without requiring any user input (e.g., tagging features).

This paper introduces L_p-Centroidal Voronoi Tessellation (L_p-CVT), a generalization of CVT well suited to anisotropic, quad-dominant and hex-dominant meshing. Like CVT, L_p-CVT is based on a standard Voronoi diagram, but it minimizes a generalized version of CVT energy that takes into account a predefined background anisotropy field and that favors cubical Voronoi cells. The L_p-CVT objective function and its gradient are computed in closed form for both surface and volume meshing (i.e., meshing a polygonal surface or meshing the interior of a polyhedral domain respectively).

After a review of previous work, the paper makes the following contributions :

- Introduce the definition of L_p-CVT and derive the objective function and its gradient in closed-form for surfaces and volumes;
- apply restricted L_2-CVT to surface remeshing. Using an anisotropy field oriented along facets normals, L_p-CVT reconstructs sharp features and self-intersections of the input surface without any need to tag them;
- apply restricted L_p-CVT to feature-sensitive quad-dominant surface remeshing and 3D L_p-CVT to hex-dominant volume meshing. To the best of our knowledge, this is the first time a variational approach is applied to hex-dominant meshing.

2 Background and previous work

Anisotropic meshing Several strategies exist for anisotropic meshing, such as "bubble packing" heuristic [Yamakawa and Shimada 2003a], tracing the curvature lines [Alliez et al. 2003], or directly minimizing the approximation error [Cohen-Steiner et al. 2004]. Labelle and Shewchuk [2003] propose a generalization of Voronoi diagrams with anisotropy attached to the vertices. Du *et al.* [2005] propose to generalize the notion of CVT with a continuous anisotropy field. They define Anisotropic CVT and experiment it in 2D. Using the same definition of anisotropy, Valette *et al.* [2008] propose a discrete approximation operating on a pre-triangulation of the domain. The latter approach shares some similarities with L_p-CVT. The differences are (1) that L_p-CVT relies on a standard Voronoi diagram (instead of an approximated anisotropic Voronoi diagram), (2) that L_p-CVT's objective function is completely determined in closed form, allowing for efficient Newton solving instead of relaxation. This makes it possible to apply L_p-CVT to more complicated settings (3D surfaces and 3D volumes) and (3) that L_p-CVT can produce quad-dominant and hex-dominant meshes.

Quad-dominant surface meshing Directly transforming and optimizing quad-meshes is studied in [Daniels-II et al. 2009]. Bubble packing, mentioned in the previous paragraph, can be adapted to optimize quads on surfaces [Itoh and Shimada 2001]. Quad-dominant mesh adaption from high-quality triangular meshes is proposed in [Tchon and Camarero 2006; Lai et al. 2008]. Using the Morse complex of a Laplacian eigenfunction defines a quadrangulation [Dong et al. 2006; Huang et al. 2008], as well as contouring the iso-u, v lines of a parameterization [Steiner and Fischer 2005; Tong et al. 2006; Kalberer et al. 2007]. Using periodic functions and mixed integer optimization avoids the need of predefining a homology basis [Ray et al. 2006; Bommes et al. 2009]. However, processing objects with sharp features still requires some constraints to be defined by the user. In contrast, L_p-CVT can reconstruct sharp features without requiring the user to tag them.

Hex-dominant volume meshing Hex-meshing is an important and difficult issue, see the survey in [Shepherd and Johnson 2008]. Several direct strategies were proposed, such as multiple sweeping [Shepherd et al. 2000], paving and plastering [Staten et al. 2005], or octree-based methods [Maréchal 2009]. As in surface meshing, it is also possible to contour a volume parameterization [Martin et al. 2008], or to use the streamsurfaces of a tensor field [Vyas and Shimada 2009]. Indirect methods first construct a tetrahedral mesh and transform it into a hex-dominant mesh, using an advancing front [Owen and Saigal 2000]. The initial tetrahedral mesh can be generated using "bubble packing" with cubic cells [Yamakawa and Shimada 2003b]. The variational hex-dominant meshing method described here uses a similar workflow, with the difference that it replaces the "bubble packing" heuristic with the optimization of an objective function (F_{L_p}) defined in closed form. This function can be more efficiently computed and the method scales-up to meshes of industrial size.

The remainder of the paper is organized as follows. The next paragraph reviews the notion of CVT (Centroidal Voronoi Tessellation). Section 3 introduces L_p-CVT, and Section 4 explains how to compute it. Section 5 demonstrates how L_p-CVT allows improving the robustness, controllability and automatic behavior of feature-sensitive surface remeshing, quad-dominant surface remeshing and variational hex-dominant meshing.

Centroidal Voronoi Tesselation Centroidal Voronoi Tessellation (CVT) is defined as the minimizer of the objective function F, referred to as the CVT energy :

$$F(\mathbf{X}) = F([\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n]) = \sum_i \int_{\Omega_i \cap \mathbf{\Omega}} \|\mathbf{y} - \mathbf{x}_i\|^2 d\mathbf{y}$$

where Ω_i denotes the 3D Voronoi cell of vertex $\mathbf{x}_i$, and $\mathbf{\Omega}$ denotes the domain to be meshed. $\mathbf{\Omega}$ can be either a surface (surface meshing), or the interior of a closed surface (volume meshing).

The gradient of F is given by [Iri et al. 1984] :

$$\nabla_F|_{\mathbf{x}_i}(\mathbf{X}) = 2m_i(\mathbf{x}_i - \mathbf{g}_i) \tag{1}$$

where m_i and $\mathbf{g}_i$ denote the volume and centroid of $\Omega_i \cap \mathbf{\Omega}$.

To minimize F, Lloyd's algorithm [1982] iteratively moves all the points $\mathbf{x}_i$ at the centroid of their Voronoi cells $\mathbf{g}_i$. To improve the speed of convergence, Du and Emelianenko [2006] solve for the fixed point of Lloyd's iteration ($\forall i, \mathbf{x}_i = \mathbf{g}_i$) using Newton's method for systems of non-linear equations . Since F is of class C^2 [Liu et al. 2009], the speed of convergence can be further improved by directly optimizing F with Newton's minimization method. To avoid costly evaluation of the Hessian at each iteration, quasi-Newton BFGS can be used instead, requiring only the evaluation of F and its gradients given in Equation 1.

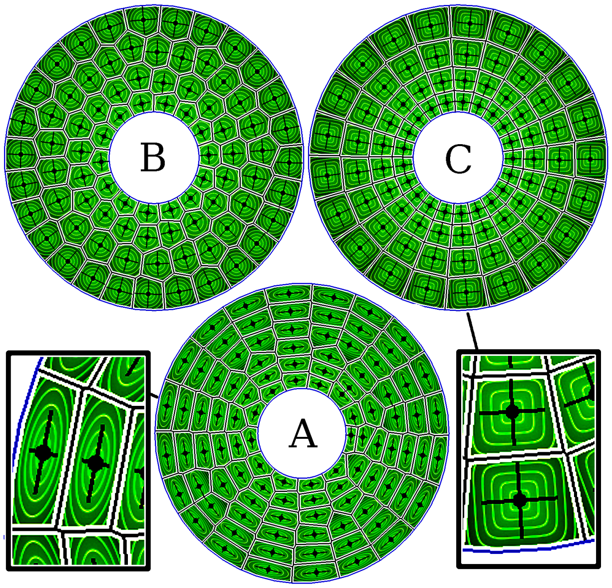


Figure 2: *CVT energy with anisotropy (A) does not take into account the axes if they have the same length (B). This motivates the definition of a new objective function (C).*

3 L_p Centroidal Voronoi Tessellation

The aim of this paper is to propose a new objective function (F_{L_p}) well suited to variational quad-dominant and hex-dominant meshing, controlled by a given anisotropy field. We make two remarks regarding the classical CVT that motivate our approach :

- First, as illustrated in Figure 2, if the axes of anisotropy have different lengths, anisotropic CVT energy can take them into account (2-A), but is not affected by them if they have the same length (2-B). This motivates generalizing CVT in a way that can take a directional-only anisotropy into account (2-C).
- Second, the "honeycomb" pattern shown in Figure 3-A is not the only critical point of CVT energy. Regular square and rectangle lattices (3-B,C) are also Voronoi diagrams that are critical points of CVT energy, since the vertices correspond to the centroids of their Voronoi cells. However, the critical points are unstable (saddles), which means that starting from (3-B) or (3-C), a small perturbation leads to a configuration similar to (3-A). As a consequence, CVT is not likely to converge to (3-B) or (3-C).

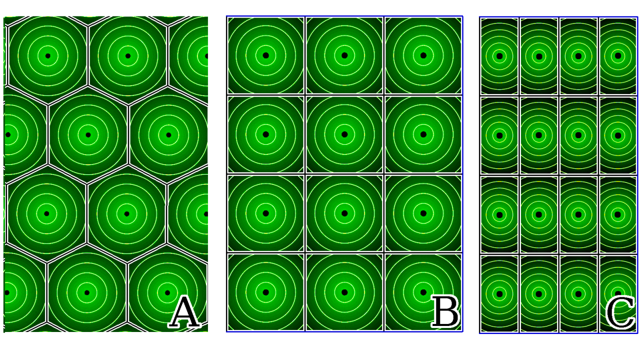
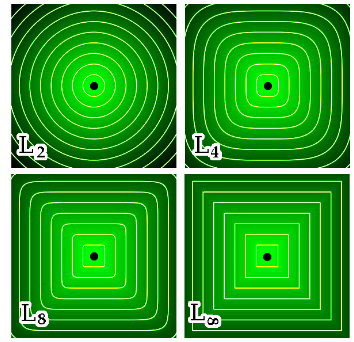


Figure 3: *A: a stable critical point of the CVT objective function F. B,C: two unstable ones. The contoured value is the L_2 norm.*



To define a variant of F that admits configuration (3-B) or (3-C) as a stable critical point, intuitively we could replace the L_2 norm with the L_∞ norm, since its iso-contours are squares (resp. cubes in 3D). However, the L_∞ norm is not differentiable. The L_p norm is a good approximation, and easier to manipulate algebraically as shown further.

3.1 Definition

L_p Centroidal Voronoi Tesselation (L_p-CVT) is defined as the minimizer of the L_p-CVT objective function F_{L_p}, obtained by injecting both the anisotropy term and the L_p norm into CVT energy :

$$F_{L_p}(\mathbf{X}) = \sum_i \int_{\Omega_i \cap \mathbf{\Omega}} \|\mathbf{M_y}[\mathbf{y} - \mathbf{x}_i]\|_p^p \, d\mathbf{y} \tag{2}$$

where $\|.\|_p$ denotes the L_p norm ($\|\mathbf{V}\|_p = \sqrt[p]{|x|^p + |y|^p + |z|^p}$ and $\|\mathbf{V}\|_p^p = |x|^p + |y|^p + |z|^p$). For an even value of p, $\|\mathbf{V}\|_p^p = x^p + y^p + z^p$. The domain $\mathbf{\Omega}$ is either a surface $\mathcal{S}$ (surface meshing, see sections 5.1,5.2,5.3), or the interior of a closed surface $\mathcal{S}$ (volume meshing, see section 5.4). An L_p Centroidal Voronoi Tessellation (L_p-CVT) is a stable critical point of F_{L_p}. The next section presents an algorithm that minimizes F_{L_p} in the general case, with $\mathcal{S}$ defined as a piecewise linear complex (PLC), for both surface and volume meshing.

If the anisotropy is defined as a symmetric tensor field $\mathbf{G_y}$, the matrix $\mathbf{M_y}$ is obtained from the SVD of $\mathbf{G_y}$: $\mathbf{M_y} = \mathbf{\Sigma}^{1/2}\mathbf{W}$ and $\mathbf{G_y} = \mathbf{W}^t\mathbf{\Sigma}\mathbf{W} = \mathbf{M_y}^t\mathbf{M_y}$ (the columns of $\mathbf{M_y}$ are the axes of the anisotropy ellipsoid).

3.2 Remark

Before explaining how to minimize F_{L_p}, it is interesting to take a close look at the relation between the gradient of the standard CVT energy F and the centroids of the Voronoi cells. Consider the integration $F_{\mathbf{\Omega}_i^0}$ of CVT energy over a *fixed* Voronoi cell $\mathbf{\Omega}_i^0$ and its gradient :

$$\begin{array}{rcl} \nabla|_{\mathbf{x}_i} F_{\mathbf{\Omega}_i^0} & = & \nabla|_{\mathbf{x}_i} \int_{\mathbf{\Omega}_i^0} \|\mathbf{y} - \mathbf{x}_i\|^2 d\mathbf{y} \\ & = & \int_{\mathbf{\Omega}_i^0} \nabla|_{\mathbf{x}_i} \|\mathbf{y} - \mathbf{x}_i\|^2 d\mathbf{y} \\ & = & 2\left(\int_{\mathbf{\Omega}_i^0} \mathbf{x}_i d\mathbf{y} - \int_{\mathbf{\Omega}_i^0} \mathbf{y} d\mathbf{y}\right) \\ & = & 2m_i(\mathbf{x}_i - \mathbf{g}_i) \end{array}$$

Detailed derivations [Iri et al. 1984] show that this result still holds when $\mathbf{\Omega}_i^0$ is replaced with a *variable* domain $\mathbf{\Omega} \cap \Omega_i$, that depends on $\mathbf{x}_i$. This is because the terms of F that depend on the variations of the integration domains $\mathbf{\Omega} \cap \Omega_i$ cancel-out when considering the influence of two neighboring Voronoi cells. This is no longer true in the anisotropic case, because of the anisotropy that varies between two adjacent cells. Therefore, to compute ∇F_{L_p}, it is not accurate to replace the centroid $\mathbf{g}_i$ and mass m_i by their anisotropic counterparts in Equation 1 since doing so ignores the variations of the domains Ω_i. In the general case, the gradient cannot be defined in terms of m_i, $\mathbf{g}_i$ and needs to be directly derived from the expression of F_{L_p}. This motivates the closed form derivations explained in the next section.

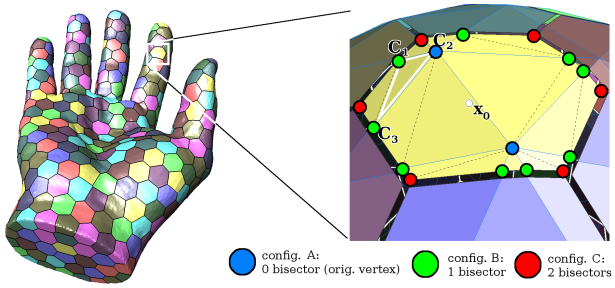


Figure 4: *Structure of the integration simplices (white triangle) for surface meshing, obtained from the restricted Voronoi diagram.*

4 Computing L_p-CVT

Given a domain $\mathbf{\Omega}$ and a set of points $\mathbf{X}$, computing L_p-CVT means minimizing the function F_{L_p}. We use BFGS, a quasi-Newton algorithm. BFGS needs to evaluate F_{L_p} and its gradient ∇F_{L_p} for a series of $\mathbf{X}$. For each iteration, the algorithm operates as follows :

- The domain $\mathbf{\Omega}$ is decomposed into a set of simplicial cells on which the expression of F_{L_p} is simple (see Section 4.1), represented by a list of integers;
- from this combinatorial representation, the value of F_{L_p} and its gradient ∇F_{L_p} is obtained in closed form (Section 4.2).

4.1 Combinatorial Structure of $F_{L_p}(\mathbf{X})$

A Piecewise Linear Complex (PLC) is a set of (possibly non-convex) polygonal facets, described by their vertices $\mathbf{p}_j$ and by their support planes $(\mathbf{N}_f, b_f)$ of equations $\mathbf{N}_f\mathbf{x} + b_f = 0$. The structure of the algorithm is the same for surface meshing (remeshing a PLC) and volume meshing (meshing the interior of a closed PLC). Determining the combinatorial structure of F_{L_p} means decomposing $\mathbf{\Omega}$ into simple cells, called *integration simplices*, on which F_{L_p} can be expressed in function of the variables $\mathbf{x}_i$ and the parameters $\mathbf{p}_j, (\mathbf{N}_f, b_f)$.

Surface meshing: The combinatorial structure of F_{L_p} is determined by the restricted Voronoi diagram (RVD), i.e. the intersection between the 3D Voronoi cells and the surface $\mathcal{S}$ (colored polygons shown in Figure 4). The RVD is first computed by an exact algorithm [Yan et al. 2009]. Then, each restricted Voronoi cell $\Omega_{\mathbf{x}_0} \cap \mathcal{S}$ is decomposed into a set of triangles. Figure 4 shows one of these triangles highlighted in white, with vertices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$;

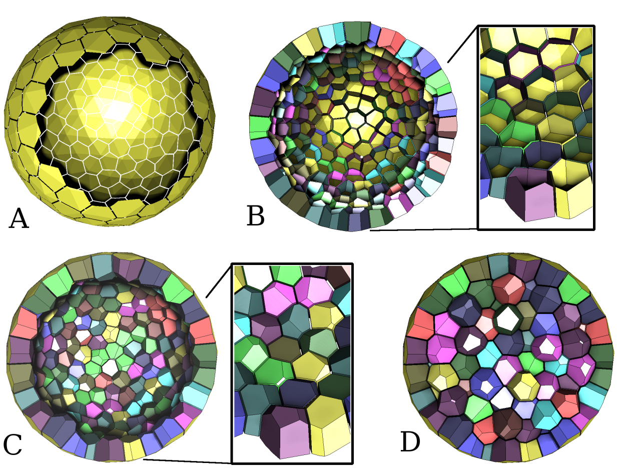


Figure 5: *Computing the clipped Voronoi diagram (see also video). Some facets/cells were removed to reveal the internal structure.*

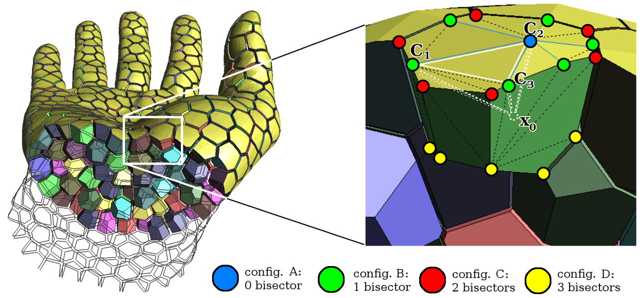
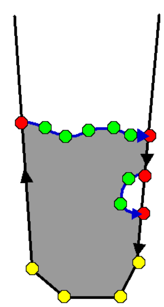


Figure 6: *Structure of the integration simplices (white tetrahedron) for volume meshing, obtained from the clipped Voronoi diagram.*

In surface meshing, the vertices $\mathbf{C}_i$ can have three different configurations, i.e., three different expressions of them in terms of the variables and parameters (see also Appendix B.2) :

- A (blue): a vertex $\mathbf{p}_j$ of $\mathcal{S}$.
- B (green): intersection of one bisector $[\mathbf{x}_0, \mathbf{x}_i]$ and two facets $(\mathbf{N}_f, b_f), (\mathbf{N}_g, b_g)$ of $\mathcal{S}$;
- C (red): intersection of two bisectors $[\mathbf{x}_0, \mathbf{x}_i], [\mathbf{x}_0, \mathbf{x}_j]$ and a facet $(\mathbf{N}_f, b_f)$ of $\mathcal{S}$;



Volume meshing: the combinatorial structure is determined by the clipped Voronoi diagram, i.e. the intersection between the 3D Voronoi cells and the interior of the closed surface $\mathcal{S}$ (colored polyhedra shown in Figure 6). These 3D clipped Voronoi cells are obtained by starting from their faces lying on S, i.e. the Restricted Voronoi Diagram (Figure 5-A). The next step computes the "walls", i.e. the clipped faces of the 3D Voronoi cells, shown in Figure 5-B. The "walls" are constructed by turning around the cells of the restricted Voronoi diagram (blue lines on the opposite illustration) and the faces of the Voronoi cells (black lines), alternating each time a vertex of configuration (C) is crossed (in red). The resulting clipped Voronoi cells are closed (Figure 5-C) by a greedy propagation over the Delaunay graph. Then, inner cells are obtained (Figure 5-D), also by a greedy propagation over the Delaunay graph. Finally, each cell $\Omega_{\mathbf{x}_0} \cap int(\mathcal{S})$ of the clipped Voronoi diagram is decomposed into tetrahedra. One of them, with vertices $\mathbf{x}_0, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$, is highlighted in white in Figure 6.

Compared with surface L_p-CVT, there is an additional possible configuration for a vertex $\mathbf{C}_i$:

- D (yellow): Voronoi vertex, determined by the intersection of three bisectors $[\mathbf{x}_0, \mathbf{x}_i], [\mathbf{x}_0, \mathbf{x}_j], [\mathbf{x}_0, \mathbf{x}_k]$;

At this stage, the combinatorial structure is determined, represented by a list of 10 integers per integration simplex $\mathbf{x}_0, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$: the first integer is the index of $\mathbf{x}_0$, and three integers per vertex $\mathbf{C}$ encode the configuration of vertex $\mathbf{C}$. A positive integer k denotes the bisector $[\mathbf{x}_0, \mathbf{x}_k]$ and a negative integer k corresponds to the boundary plane $(\mathbf{N}_{-k}, b_{-k})$. The implementation is provided in `LpCVT/combinatorics`, in the supplemental material.

4.2 Algebraic Structure of $F_{L_p}(\mathbf{X})$

The energy F_{L_p} and its gradient ∇F_{L_p} are computed from the combinatorial structure (array of integers), the array of vertices $(\mathbf{x}_i)$ and the array of boundary planes $(\mathbf{N}_f, b_f)$, by summing the contributions $F^T_{L_p}$ and $\nabla F^T_{L_p}$ of each integration simplex T. The remainder of this section and the Appendix detail this computation.

Expression of F_{L_p}

The integration $F_{L_p}^T$ of the L_p energy over an integration simplex T is given by :

$$\begin{array}{rcl} F_{L_p}^T & = & \int\limits_T \|\mathbf{M}_T(\mathbf{y}-\mathbf{x}_0)\|_p^p \, d\mathbf{y} \\ & = & \frac{|T|}{\binom{n+p}{n}} \sum\limits_{\alpha+\beta+\gamma=p} \overline{\mathbf{U}_1^\alpha * \mathbf{U}_2^\beta * \mathbf{U}_3^\gamma} \end{array} \tag{3}$$

$$\text{where: } \left\{ \begin{array}{lcl} \mathbf{U_i} & = & \mathbf{M}_T(\mathbf{C}_i - \mathbf{x}_0) \\ \mathbf{V}_1 * \mathbf{V}_2 & = & [x_1x_2, y_1y_2, z_1z_2]^t \\ \mathbf{V}^\alpha & = & \mathbf{V} * \mathbf{V} * \ldots * \mathbf{V} (\alpha \text{ times}) \\ \overline{\mathbf{V}} & = & x+y+z \end{array} \right.$$

In surface meshing, T denotes the triangle $T(\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3)$ and $n = 2$. In volume meshing, T denotes the tetrahedron $T(\mathbf{x}_0, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3)$ and $n = 3$.

Proof: see Appendix A.

Expression of ∇F_{L_p}

The gradient of $F_{L_p}^T$ relative to $\mathbf{X}$ is obtained by the chain rule :

$$\frac{dF_{L_p}^T(\mathbf{x}_0, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3)}{d\mathbf{X}} = \frac{dF_{L_p}^T}{d\mathbf{x}_0} + \frac{dF_{L_p}^T}{d\mathbf{C}_1}\frac{d\mathbf{C}_1}{d\mathbf{X}} + \frac{dF_{L_p}^T}{d\mathbf{C}_2}\frac{d\mathbf{C}_2}{d\mathbf{X}} + \frac{dF_{L_p}^T}{d\mathbf{C}_3}\frac{d\mathbf{C}_3}{d\mathbf{X}}$$

where $d\mathbf{A}/d\mathbf{B} = \left({\partial a_i}/{\partial b_j} \right)_{i,j}$ denotes the Jacobian matrix of $\mathbf{A}$ with respect to $\mathbf{B}$. See Appendix B for the detailed derivations. The multithreaded implementation is provided in `LpCVT/algebra`, in the supplemental material. The thorough analysis of the continuity of F_{L_p} will be addressed in a future publication. Experimentally, BFGS applied to F_{L_p} behaves well (does not oscillate).

5 Applications

We shall now present some applications to surface remeshing and volume meshing. All tests were performed on a 8-cores, 2.2 GHz computer running the multithreaded implementation (Section 4).

5.1 Anisotropic Surface Remeshing

The simplest application of L_p-CVT is anisotropic surface meshing ($p = 2$). As a generalization of [Yan et al. 2009] to the anisotropic setting, anisotropic L_2-CVT can be used to remesh a surface with prescribed anisotropy. The surface meshing framework is used, i.e. F_{L_p} is minimized on the restricted Voronoi diagram. Each facet f has an associated anisotropy $\mathbf{M}^f$, defined as a smoothed estimate of the curvature tensor [Ray et al. 2006]. Typical examples are shown in Figures 7 and 1.

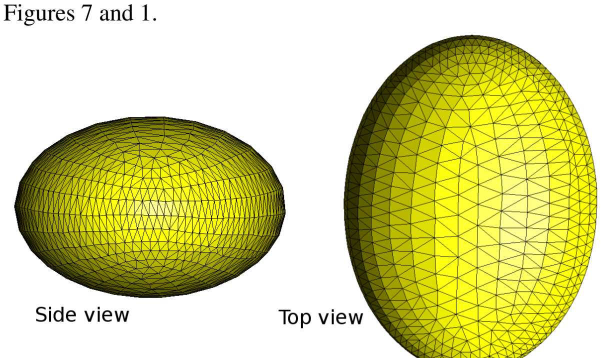


Figure 7: *Remeshing an ellipsoid with anisotropic L_2-CVT. Vertices are aligned along the principal curvature lines.*

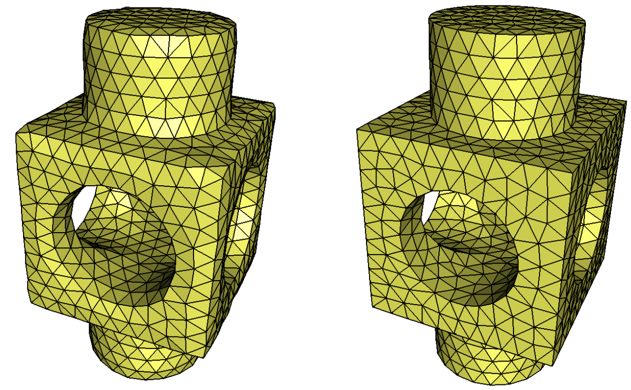


Figure 8: *Remeshing the 'block' dataset. Left: with standard CVT energy. Right: L_2-CVT with normal anisotropy recovers the features, without needing pre-tagging or constrained optimization.*

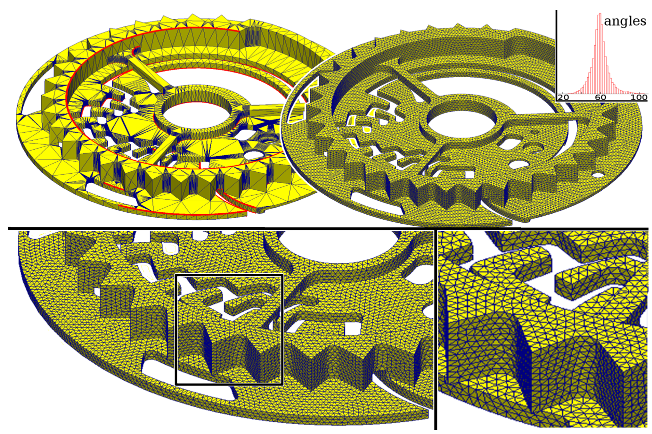


Figure 9: *Left: 'mazewheel' dataset (borders shown in red); Right: isotropic remesh and angles histogram. Timing: 132 seconds.*

5.2 Fully Automatic Feature-Sensitive Remeshing

Remeshing surfaces with features is a challenging problem. Existing approaches [Tournois et al. 2009; Yan et al. 2009] let the user specify which edges correspond to features, and use constrained optimization to sample them properly. For mechanical parts, it may be tedious to select the feature edges by hand (see e.g. Figure 9). With a specific definition of per-facet normal anisotropy, the L_p-CVT objective function naturally recovers the features, without needing to tag them or using any expensive constrained optimization. The normal anisotropy $\mathbf{M}^f$ associated with facet f is defined as follows :

$$\mathbf{M}^f = (s-1)\begin{pmatrix} \mathbf{N}_x^f[\mathbf{N}^f]^t \\ \mathbf{N}_y^f[\mathbf{N}^f]^t \\ \mathbf{N}_z^f[\mathbf{N}^f]^t \end{pmatrix} + \mathbf{I}_{3\times3}$$

where $\mathbf{N}^f$ denotes the unit normal of facet f and s the importance of normal anisotropy ($s = 5$ in the results herein). Applying $\mathbf{M}^f$ to a vector $\mathbf{v}$ magnifies the component of $\mathbf{v}$ aligned with $\mathbf{N}^f$ by a factor of s. In other words, normal anisotropy penalizes the vertices that are far away from the tangent plane of the surface. On a sharp edge e, the combined effects of the anisotropies of both facets incident to e tend to attract vertices onto e. Figures 8,9,10 show some examples, statistics and timings.

To ensure the remesh is homeomorphic to the initial surface, it is sufficient to satisfy the Topological Ball Property [Edelsbrun-

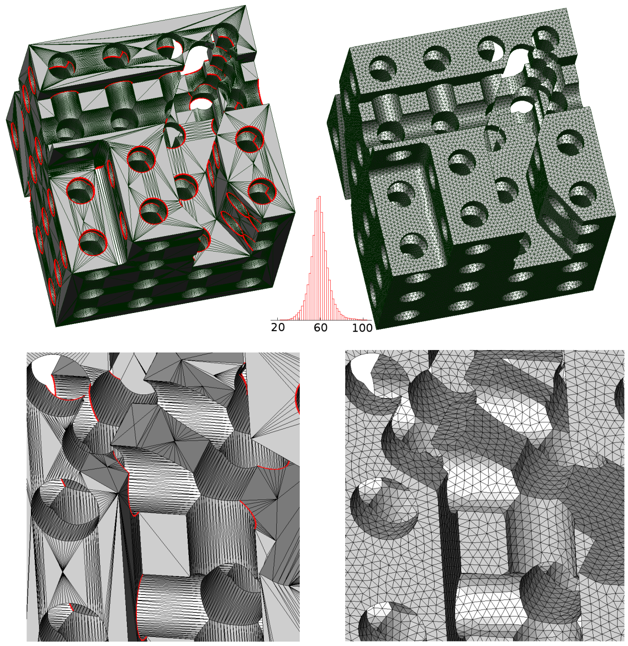


Figure 10: *Left: 'nastycheese' dataset (borders shown in red); Right: isotropic remesh and angles histogram. Bottom: closeup. Sharp features are naturally reconstructed by L_2-CVT with normal anisotropy, without needing to detect or tag them. Timing: 215 s.*

ner and Shah 1997], i.e. each restricted Voronoi vertex/edge/face should be homeomorphic to a point/segment/disc. This property is a special case of the nerve theorem, also valid for any partition of the original surface. For restricted Voronoi cells that have multiple components, we consider each component as an independent cell and compute the dual of the so-defined partition. Considering the connected components as individual cells properly recovers thin features. After separating the connected components, if some restricted Voronoi faces are still not homeomorphic to disc (e.g. cylinders), topology control [Yan et al. 2009] is performed, i.e. interleaved Delaunay refinement and optimization of F_{L_2}. Figure 11 shows a detail of the 'nastycheese' surface, with such non-manifold configurations resolved by the method.

Configurations with non-conforming, self-intersecting surfaces can also be handled by normal anisotropy. The intersections curves do not need to be computed explicitly, they are naturally reconstructed by the L_2-CVT framework with normal anisotropy, for the same reason as sharp features mentioned above. On an intersection, the combined anisotropies of both sheets place vertices on the intersection curve, that appears in the remesh as non-manifold edges, shown in red in Figure 12. A CAD dataset is shown in Figure 12-bottom. The remeshing framework fixes the gaps (artificial borders shown in red) and samples intersection curves without any pre-processing or manual intervention.

Figure 13 compares the result with the method developed by Dey *et al.* [2008] based on Delaunay refinement and protecting balls. Delaunay refinement has the advantage of providing a solid mathematical foundation that helps proving that the remesh is isotopic to the original surface, and that the pre-determined creases are recovered. In our case, using the connected components of the RVD and topology control ensures that the remesh is homeomorphic to the initial surface. As a variational method, L_2-CVT globally optimizes the vertices locations, resulting in nearly equilateral triangles of homogeneous sizes.

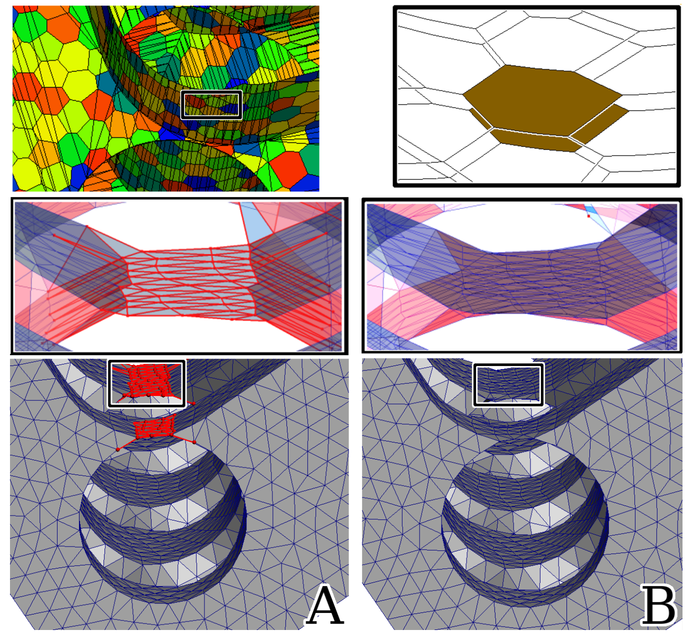


Figure 11: *Thin features create restricted Voronoi cells with several connected components. A: this results in sheets of non-manifold triangles; B: the dual of the connected components of the RVD recovers the correct topology, with two sheets of triangles.*

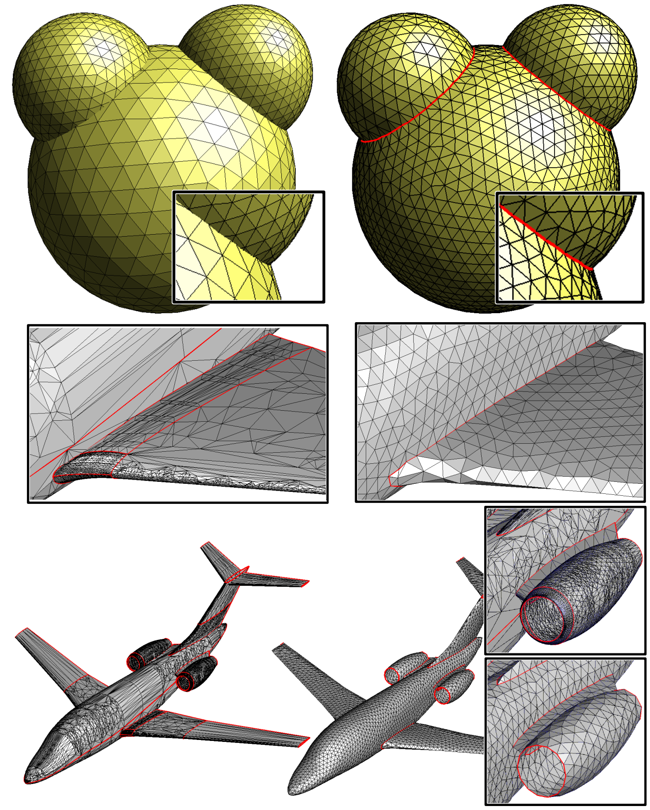


Figure 12: *Remeshing surfaces with self-intersections. L_2-CVT with normal anisotropy naturally samples the intersections.*

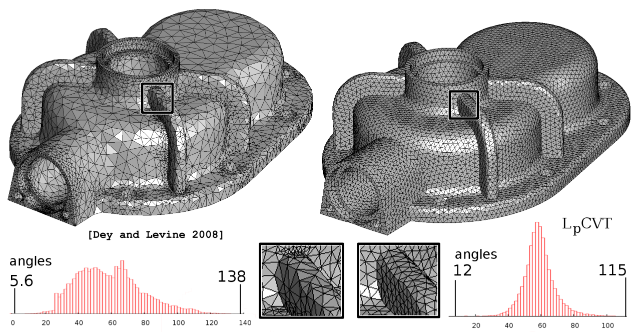


Figure 13: *Comparison with Delaunay refinement [Dey and Levine 2008] (data kindly provided by paper's author). L_2-CVT timing: 236 seconds (8 cores, 2.26 GHz).*

However, since creases are not tagged, we cannot prove that anisotropic L_2-CVT faithfully recovers them in all possible cases. To show the robustness and practical value of L_2-CVT, further tests are provided in the supplemental material (images and meshes) on a variety of meshes with unconformities (artificial borders, creases ...). Studying how L_2-CVT behaves from the point of view of ε-sampling generalized to non-smooth surfaces [Boissonnat and Oudot 2006] may yield theoretical guarantees, but this may be difficult to achieve with a variational method that moves all the vertices.

5.3 Variational Quad-Dominant Surface Remeshing

As discussed in Section 3, L_p-CVT can be applied to quad-dominant surface meshing. Using a value of p that is sufficiently large gives a good approximation of the L_∞ norm ($p = 8$ in the results herein). Figure 14 shows an example of quad-dominant remeshing, using L_8-CVT. An estimate of the principal directions of curvatures [Ray et al. 2006] is used to steer the orientation of the quads. The normal vector is scaled by 5, to ensure that sharp features are reconstructed (see previous subsection). We perform interleaved refinement / optimization of F_{L_p} iterations, as done in [Tournois et al. 2009] :

(1) distribute vertices randomly
(2) optimize F_{L_8}
(3) for each refinement iteration
  (4) insert a new vertex at the center of each edge of the RDT
  (5) optimize F_{L_8}
(6) compute the Restricted Delaunay Triangulation
(7) merge triangles in priority order

Refinement for quad meshes (step 4) consists in inserting a vertex in the middle of each edge of the restricted Delaunay triangulation. Finally, the quad-dominant remesh is extracted from the restricted Delaunay triangulation (Figure 14 top left) by merging pairs of triangles, sorted by the angle at the corners of the so-obtained quads (Figure 14 top right). The result is compared with [Ray et al. 2006] and the most recent [Bommes et al. 2009] quad-remeshing methods. The strong point of [Bommes et al. 2009] is the ability of generating a nearly regular quad-only mesh, well suited to Splines fitting. The advantages of L_p-CVT are the fact that it is fully automatic, and the smaller variations in angles and edge lengths, suitable for numerical simulations that rely on homogeneous element sizes.

Besides orientation of the elements, the anisotropy matrix $\mathbf{M}^f$ associated with each facet f can also prescribe element sizes (by scaling the axes, i.e., the columns of $\mathbf{M}^f$), as shown in Figure 15. In this example, an approximation of local feature size is used to generate smaller elements in curved and thin zones. If a different scaling factor is applied to each anisotropy axis, then an anisotropic rectangle/triangle mesh is obtained (see Figure 16). In this example, the used anisotropy is an approximation of the curvature tensor.

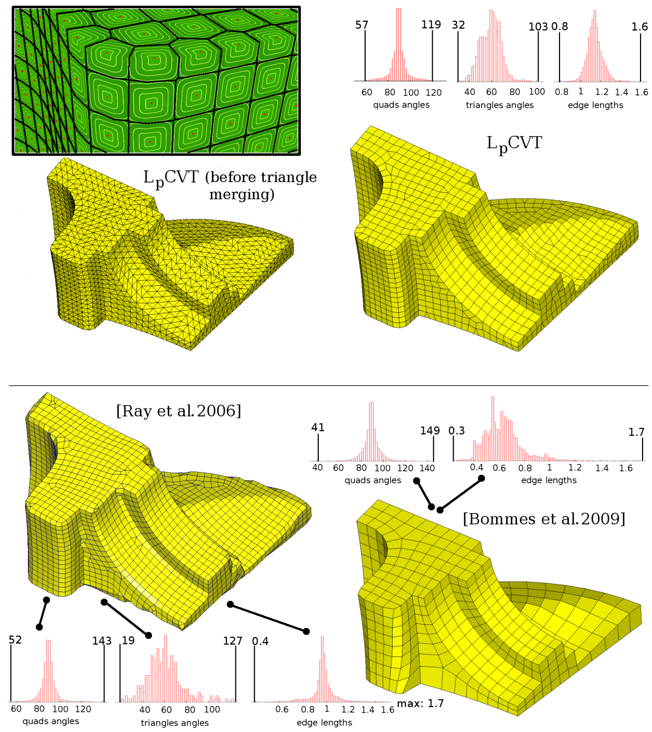


Figure 14: *Variational Quad-Dominant Remeshing of the 'fandisk' dataset and comparison with previous work (data kindly provided by paper's author). L_p-CVT timing: 322 seconds.*

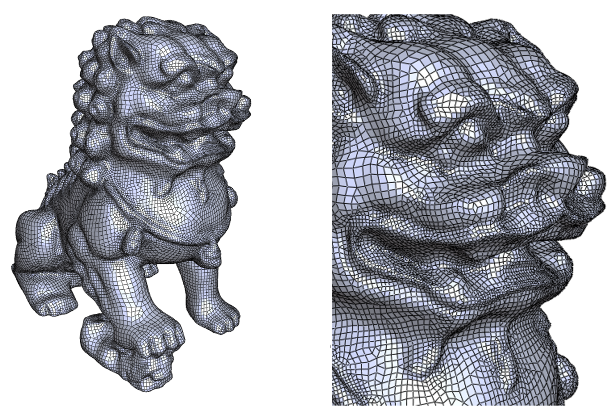


Figure 15: *Variational Quad-Dominant Remeshing of the 'chinese lion' dataset with varying element sizes. Timing: 412 seconds.*

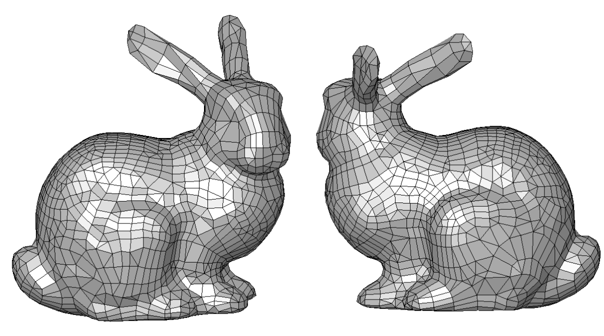


Figure 16: *Variational Quad-Dominant Remeshing of the 'Stanford bunny' dataset with anisotropic elements. Timing: 271 seconds.*

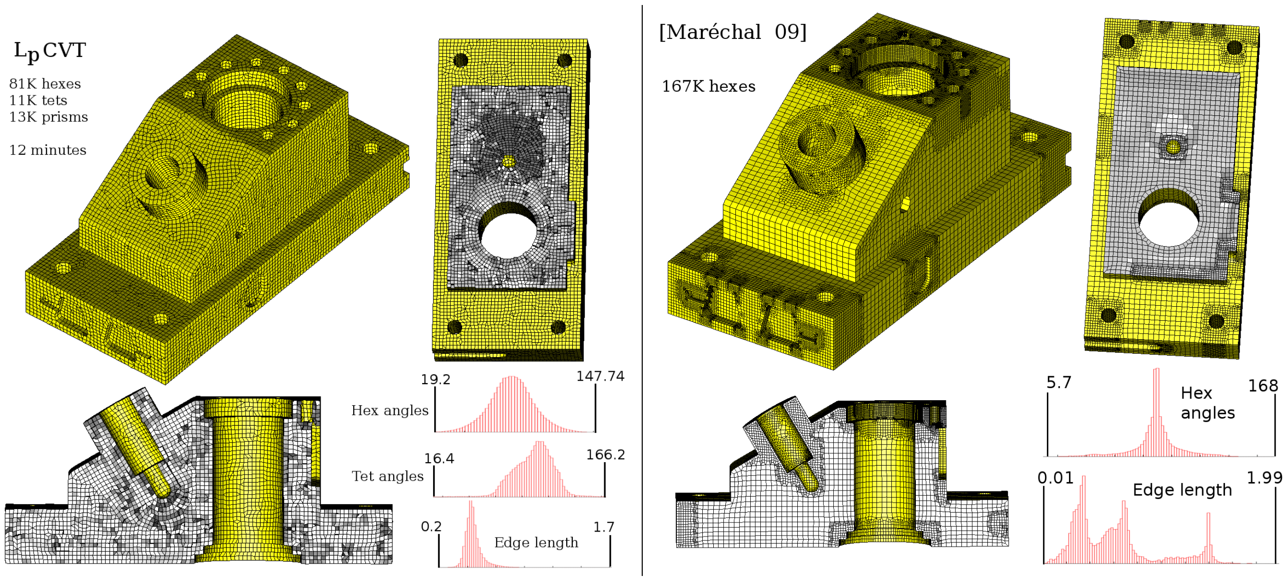
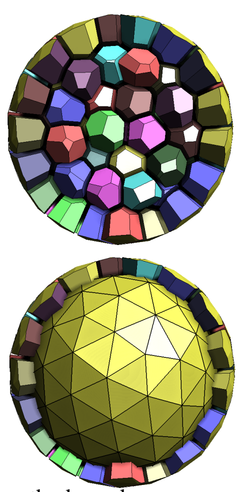


Figure 17: *Variational Hex-Dominant Meshing and comparison with [Maréchal 2009] (data kindly provided by paper's author).*

5.4 Variational Hex-Dominant Meshing



Our framework extends naturally to variational hex-dominant meshing. The volume framework is used (i.e., L_p-CVT is computed on the clipped Voronoi diagram), with an anisotropy matrix $\mathbf{M}$ associated with each integration simplex. The anisotropy is attached to the facets of the surface, defined as in the previous section, and each integration simplex uses the anisotropy $\mathbf{M}^f$ of the facet f nearest to its centroid, queried using Approximated Nearest Neighbor [Mount and Arya 1997]. Note that the Delaunay triangulation restricted to the interior of $\mathcal{S}$ is smaller than the interior of $\mathcal{S}$, as shown in the opposite figure. To avoid this "shrinkage", vertices on the boundary require a special treatment, similarly to [Tournois et al. 2009]. Vertices on the boundary are computed by the surface framework, as in the previous subsection, using the same values of $\mathbf{M}^f$. These vertices are characterized by a non-empty intersection of their Voronoi cells with the surface $\mathcal{S}$. After the optimization step, the algorithm computes the Delaunay triangulation of the vertices restricted to the interior of $\mathcal{S}$, using the combinatorial information computed in Section 4.1. Each vertex of configuration D (Voronoi vertex inside $\mathcal{S}$) corresponds to a tetrahedron. Finally, the tetrahedra are merged to form hexes, using combinatorial optimization [Meshkat and Talmor 2000]. The subgraphs of the simplex graph that correspond to hexes are pattern-matched and hexes are generated, in priority order, determined by dihedral angles and face planarity.

Figure 17 shows a hex-dominant mesh (boundary and cross-sections) of a CAD model with timing and stats, and compares the result with an octree-based method [Maréchal 2009]. The advantage of the octree-based method are that it generates a pure hexahedral mesh with a nearly regular pattern and that it is fast. The advantages of L_p-CVT is that boundaries are well respected regardless their orientation, and that it generates homogeneous element sizes. Figures 1 and 19 show examples with smooth surfaces.

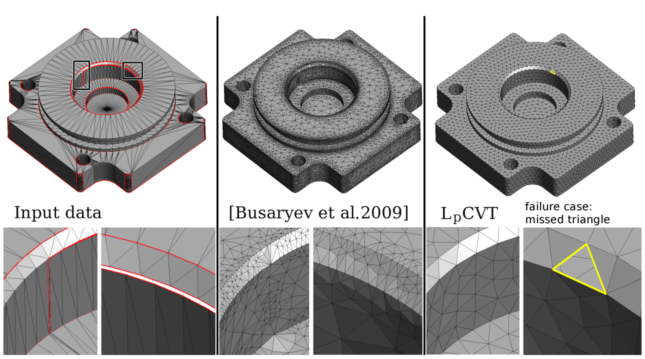


Figure 18: *Mesh repair, comparison with Delaunay refinement [Busaryev et al. 2009], (data kindly provided by paper's author).*

5.5 Discussion - Limitations

L_p-CVT is robust to T-junctions and artificial borders (see the empirical results in the paper and supplemental material), i.e. meshes that have a correct geometry but a non-conforming discretization. However, L_p-CVT might not handle properly models that are geometrically inconsistent. Figure 18 compares the result with the CAD model repair algorithm proposed in [Busaryev et al. 2009] and illustrates this limitation of L_p-CVT. The input data has some gaps. L_p-CVT misses triangles when a Voronoi edge passes through a gap (more triangles are missed when the number of vertices is increased). In practice, hole filling heuristic fixes the problem in most cases. More formally, we think that the ε-sampling formalism and its extensions [Boissonnat and Oudot 2006] provide a mean of developing the theoretical tools to study how to recover the missing triangles with theoretical guarantees. Note also that the continuity of F_{L_p} remains to be studied. Finally, we also mention that under extreme anisotropy (100x ratio between the lengths of axes) convergence becomes much slower because of the conditioning of the Hessian of F_{L_p}.

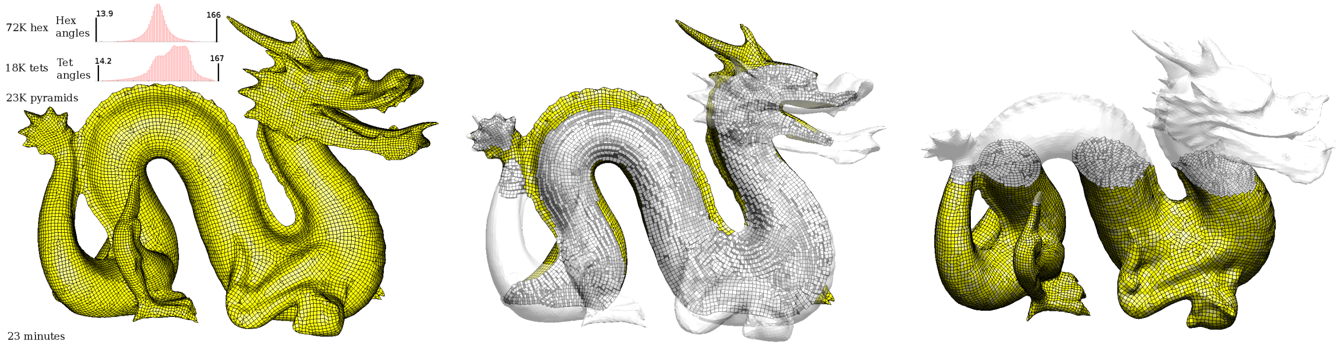


Figure 19: *Variational Hex-Dominant Meshing of the 'dragon' dataset. Timing: 23 minutes.*

Acknowledgments

This project is supported by the European Research Council (GOODSHAPE ERC-StG-205693). The authors want to warmly thank Rhaleb Zayer and Wenping Wang for many discussions, Jean-François Remacle for help with GMSH, Nicolas Saugnier for parallelization, Dong-Ming Yan for the implementation of RVD, the CGAL project for Delaunay triangulation, Loic Marechal and Pierre alliez for data and discussion, Tamal Dey for data and help with DelPSC, David Bommes, Leif Kobbelt and Aim@Shape for data, and the anonymous reviewers for their suggestions that helped improving the paper.

A Expression of F_{L_p}

$$
\begin{aligned}
F^T_{L_p} &= \int_{T(\mathbf{x}_0,\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3)} \|\mathbf{M}_T(\mathbf{y}-\mathbf{x}_0)\|_p^p \, d\mathbf{y} \\
&= \frac{|T|}{\binom{n+p}{n}} \underbrace{\sum_{\alpha+\beta+\gamma=p} \overline{\mathbf{U}_1^\alpha * \mathbf{U}_2^\beta * \mathbf{U}_3^\gamma}}_{E^T_{L_p}}
\end{aligned} \tag{4}
$$

$$
\text{where:} \quad \left\{
\begin{array}{lcl}
\mathbf{U_i} & = & \mathbf{M}_T(\mathbf{C}_i - \mathbf{x}_0) \\
\mathbf{V}_1 * \mathbf{V}_2 & = & [x_1 x_2, y_1 y_2, z_1 z_2]^t \\
\mathbf{V}^\alpha & = & \mathbf{V} * \mathbf{V} * \ldots * \mathbf{V} (\alpha \text{ times}) \\
\overline{\mathbf{V}} & = & x + y + z
\end{array}
\right.
$$

Proof:

The change of variable $\mathbf{x} := \mathbf{M}_T(\mathbf{y} - \mathbf{x}_0)$ gives :

$$
F^T_{L_p} = \int_{T(0,\mathbf{U}_1,\mathbf{U}_2\mathbf{U}_3)} \overline{\mathbf{x}^p} d\mathbf{x}
$$

where the integrand is a homogeneous polynomial. The integration of a homogeneous polynomial of degree p over a simplex Δ of dimension n [Lasserre and Avrachenkov 2001] is given by :

$$
\int_{\Delta_n} H_p(\mathbf{x}, \mathbf{x} \ldots \mathbf{x}) = \frac{|\Delta|}{\binom{n+p}{n}} \sum_{0 \le i_1 \le i_2 \ldots \le i_p \le n} H_p(\mathbf{x}_{i_1}, \mathbf{x}_{i_2} \ldots \mathbf{x}_{i_p})
$$

where H_p denotes the polar form of the polynomial. Substituting H_p with $\overline{\mathbf{x_1} * \mathbf{x_2} * \ldots * \mathbf{x}_p}$ gives the result. □

B Expression of ∇F

The gradient of F is obtained by combining the gradient of F relative to the vertices of the integration simplices (Appendix B.1) with the gradients of Voronoi vertices (Appendix B.2) using the chain rule.

B.1 Expression of $dF^T_{L_p}$

We first derive $\frac{dF^T_{L_p}}{d\mathbf{x}_0\mathbf{U}_1\mathbf{U}_2\mathbf{U}_3}$ and then $\frac{d\mathbf{C}_i}{d\mathbf{X}}$. Recalling that $F^T_{L_p} = |T| E^T_{L_p}$ where $E^T_{L_p}$ is given Equation 4 in Appendix A, we obtain :

$$
dF^T_{L_p} = d(|T| \, E^T_{L_p}) = E^T_{L_p} d|T| + |T| dE^T_{L_p}
$$

$$
\frac{dE^T_{L_p}}{d\mathbf{U}_1\mathbf{U}_2\mathbf{U}_3} = \begin{bmatrix}
\sum_{\alpha+\beta+\gamma=p;\alpha\ge 1} \alpha U_1^{\alpha-1} * U_2^\beta * U_3^\gamma \\
\sum_{\alpha+\beta+\gamma=p;\beta\ge 1} \beta U_1^{\alpha} * U_2^{\beta-1} * U_3^\gamma \\
\sum_{\alpha+\beta+\gamma=p;\gamma\ge 1} \gamma U_1^{\alpha} * U_2^\beta * U_3^{\gamma-1}
\end{bmatrix}^t
$$

In surface meshing, $|T| = 1/2 \, \|\mathbf{N}\|$ and :

$$
\frac{d|T|}{d\mathbf{U_1 U_2 U_3}} = \frac{-1}{2|T|} \left([\mathbf{N} \times (\mathbf{U}_2 - \mathbf{U}_3)]^t \, [\mathbf{N} \times (\mathbf{U}_3 - \mathbf{U}_1)]^t \, [\mathbf{N} \times (\mathbf{U}_1 - \mathbf{U}_2)]^t \right)
$$

In volume meshing, $|T| = 1/6 \, \mathbf{U_1} \cdot (\mathbf{U_2} \times \mathbf{U_3})$, and :

$$
\frac{d|T|}{d\mathbf{U}_1\mathbf{U}_2\mathbf{U}_3} = 1/6 \, \left([\mathbf{U}_2 \times \mathbf{U}_3]^t \, [\mathbf{U}_3 \times \mathbf{U}_1]^t \, [\mathbf{U}_1 \times \mathbf{U}_2]^t \right)
$$

Finally, the derivatives of $F^T_{L_p}$ with respect to $\mathbf{x}_0, \mathbf{C}_1, \mathbf{C}_2$ and $\mathbf{C}_3$ are given by :

$$
\frac{dF^T_{L_p}}{d\mathbf{C}_i} = \frac{dF^T_{L_p}}{d\mathbf{U}_i}\mathbf{M}_T \quad ; \quad \frac{dF^T_{L_p}}{d\mathbf{x}_0} = -\frac{dF^T_{L_p}}{d\mathbf{C}_1} - \frac{dF^T_{L_p}}{d\mathbf{C}_2} - \frac{dF^T_{L_p}}{d\mathbf{C}_3}
$$

B.2 Gradients of Voronoi vertices

The gradient of a circumcenter $\mathbf{C}$ relative to the four vertices of the tetrahedron is given by :

config. B: 1 bisector $[\mathbf{x}_0\mathbf{x}_1]$ and two planes $(\mathbf{N}_1, b_1), (\mathbf{N}_2, b_2)$

$$d\mathbf{C} = \begin{pmatrix} [\mathbf{x}_1-\mathbf{x}_0]^t \\ [\mathbf{N}_1]^t \\ [\mathbf{N}_2]^t \end{pmatrix}^{-1} \begin{pmatrix} [\mathbf{C}-\mathbf{x}_0]^t & [\mathbf{x}_1-\mathbf{C}]^t \\ [\mathbf{C}-\mathbf{x}_0]^t & 0 \\ 0 & 0 \end{pmatrix}$$

config. C: 2 bisectors $[\mathbf{x}_0\mathbf{x}_1], [\mathbf{x}_0\mathbf{x}_2]$ and 1 plane $(\mathbf{N}_1, b_1)$

$$d\mathbf{C} = \begin{pmatrix} [\mathbf{x}_1-\mathbf{x}_0]^t \\ [\mathbf{x}_2-\mathbf{x}_0]^t \\ [\mathbf{N}_1]^t \end{pmatrix}^{-1} \begin{pmatrix} [\mathbf{C}-\mathbf{x}_0]^t & [\mathbf{x}_1-\mathbf{C}]^t & 0 \\ [\mathbf{C}-\mathbf{x}_0]^t & 0 & [\mathbf{x}_2-\mathbf{C}]^t \\ 0 & 0 & 0 \end{pmatrix}$$

config. D: 3 bisectors $[\mathbf{x}_0\mathbf{x}_1], [\mathbf{x}_0\mathbf{x}_2], [\mathbf{x}_0\mathbf{x}_3]$

$$d\mathbf{C} = \begin{pmatrix} [\mathbf{x}_1-\mathbf{x}_0]^t \\ [\mathbf{x}_2-\mathbf{x}_0]^t \\ [\mathbf{x}_3-\mathbf{x}_0]^t \end{pmatrix}^{-1} \begin{pmatrix} [\mathbf{C}-\mathbf{x}_0]^t & [\mathbf{x}_1-\mathbf{C}]^t & 0 & 0 \\ [\mathbf{C}-\mathbf{x}_0]^t & 0 & [\mathbf{x}_2-\mathbf{C}]^t & 0 \\ [\mathbf{C}-\mathbf{x}_0]^t & 0 & 0 & [\mathbf{x}_3-\mathbf{C}]^t \end{pmatrix}$$

Proof:

We show the result for configuration D (circumcenter of Delaunay simplex). The result for configurations B and C is obtained similarly. Configuration A, i.e. original vertex of $\mathcal{S}$, yields no derivative since it does not depend on the $\mathbf{x}_i$s. In configuration D, the point $\mathbf{C}$ is the circumcenter of the Delaunay simplex $T(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, obtained by intersecting three bisectors $[\mathbf{x}_0\mathbf{x}_1]$, $[\mathbf{x}_0\mathbf{x}_2]$, $[\mathbf{x}_0\mathbf{x}_3]$:

$$\mathbf{C} = \mathbf{A}^{-1}\mathbf{B} \quad \text{where} \quad \mathbf{A} = \begin{pmatrix} [\mathbf{x}_1 - \mathbf{x}_0]^t \\ [\mathbf{x}_2 - \mathbf{x}_0]^t \\ [\mathbf{x}_3 - \mathbf{x}_0]^t \end{pmatrix} \quad ; \quad \mathbf{B} = \frac{1}{2}\begin{pmatrix} \mathbf{x}_1^2 - \mathbf{x}_0^2 \\ \mathbf{x}_2^2 - \mathbf{x}_0^2 \\ \mathbf{x}_3^2 - \mathbf{x}_0^2 \end{pmatrix}$$

We first recall some matrix derivation rules [Minka 1997]:

$$\begin{array}{lrcl} (1) & d(\mathbf{AB}) & = & d\mathbf{AB} + \mathbf{A}d\mathbf{B} \\ (2) & d(\mathbf{A}^{-1}) & = & -\mathbf{A}^{-1}(d\mathbf{A})\mathbf{A}^{-1} \end{array}$$

Using first (1) then (2), the expression of $d\mathbf{C}$ can be expanded :

$$\begin{array}{rcrcl} d\mathbf{C} & = & d(\mathbf{A}^{-1}\mathbf{B}) & = & (d\mathbf{A}^{-1})\mathbf{B} + \mathbf{A}^{-1}d\mathbf{B} \\ & & & = & -\mathbf{A}^{-1}(d\mathbf{A})\mathbf{A}^{-1}\mathbf{B} + \mathbf{A}^{-1}d\mathbf{B} \\ & & & = & \mathbf{A}^{-1}(d\mathbf{B} - (d\mathbf{A})\mathbf{C}) \end{array}$$

then $(d\mathbf{B} - (d\mathbf{A})\mathbf{C})$ is substituted with :

$$d\mathbf{B} = \begin{pmatrix} -\mathbf{x}_0^t & \mathbf{x}_1^t & 0 & 0 \\ -\mathbf{x}_0^t & 0 & \mathbf{x}_2^t & 0 \\ -\mathbf{x}_0^t & 0 & 0 & \mathbf{x}_3^t \end{pmatrix} \quad ; \quad (d\mathbf{A})\mathbf{C} = \begin{pmatrix} -\mathbf{C}^t & \mathbf{C}^t & 0 & 0 \\ -\mathbf{C}^t & 0 & \mathbf{C}^t & 0 \\ -\mathbf{C}^t & 0 & 0 & \mathbf{C}^t \end{pmatrix}$$

which gives the result. □

The equations for configurations C and B are obtained by replacing the equations of the last (resp. the last two) bisectors with the equations of a constant plane $\mathbf{N}_1^t\mathbf{C}+b_1 = 0$ (resp. $\mathbf{N}_2^t\mathbf{C}+b_2 = 0$).

References

ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LÉVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Transactions on Graphics 22*, 3, 485–493.

ALLIEZ, P., COHEN-STEINER, D., YVINEC, M., AND DESBRUN, M. 2005. Variational tetrahedral meshing. *ACM Transactions on Graphics 24*, 3, 617–625.

BOISSONNAT, J.-D., AND OUDOT, S. 2006. Provably good sampling and meshing of Lipschitz surfaces. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, 337–346.

BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. *ACM Transactions on Graphics 28*, 3, 1–10.

BUSARYEV, O., DEY, T. K., AND LEVINE, J. A. 2009. Repairing and meshing imperfect shapes with Delaunay refinement. In *SPM '09: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, 25–33.

CHEN, L., AND XU, J. 2004. Optimal Delaunay triangulations. *Journal of Computational Mathematics 22*, 2, 299–308.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. *ACM Transactions on Graphics 23*, 905–914.

DANIELS-II, J., SILVA, C. T., AND COHEN, E. 2009. Localized quadrilateral coarsening. *Computer Graphics Forum 28*, 1437–1444.

DEY, T. K., AND LEVINE, J. A. 2008. Delaunay meshing of piecewise smooth complexes without expensive predicates. *Algorithms 2*, 1327–1349. OSU-CISRC-7/08-TR40.

DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. *ACM Transactions on Graphics 25*, 1057–1066.

DU, Q., AND EMELIANENKO, M. 2006. Acceleration schemes for computing centroidal Voronoi tessellations. *Numerical Linear Algebra with Applications 13*, 2–3, 173–192.

DU, Q., AND WANG, D. 2003. Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations. *International Journal for Numerical Methods in Engineering 56*, 1355–1373.

DU, Q., AND WANG, D. 2005. Anisotropic centroidal Voronoi tessellations and their applications. *SIAM Journal on Scientific Computing 26*, 3, 737–761.

DU, Q., FABER, V., AND GUNZBURGER, M. 1999. Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review 41*, 4, 637–676.

EDELSBRUNNER, H., AND SHAH, N. R. 1997. Triangulating topological spaces. *International Journal of Computational Geometry & Applications 7*, 4, 365–378.

HUANG, J., ZHANG, M., MA, J., LIU, X., KOBBELT, L., AND BAO, H. 2008. Spectral quadrangulation with orientation and alignment control. *ACM Transactions on Graphics 27*.

IRI, M., MUROTA, K., AND OHYA, T. 1984. A fast Voronoi-diagram algorithm with applications to geographical optimization problems. In *Proc. IFIP*, 273–288.

ITOH, T., AND SHIMADA, K. 2001. Automatic conversion of triangular meshes into quadrilateral meshes with directionality. *International Journal of CAD/CAM*.

KALBERER, F., NIESER, M., AND POLTHIER, K. 2007. Quadcover — surface parameterization using branched coverings. *Computer Graphics Forum 26*, 375–384.

LABELLE, F., AND SHEWCHUK, J. R. 2003. Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, 191–200.

LAI, Y.-K., KOBBELT, L., AND HU, S.-M. 2008. An incremental approach to feature aligned quad dominant remeshing. In *ACM SPM conference proceedings*.

LASSERRE, J., AND AVRACHENKOV, K. 2001. Integration on a simplex: The multi-dimensional version of $\int_a^b x^p dx$. *American Mathematics Monthly 108*, 151–154.

LIU, Y., WANG, W., LÉVY, B., SUN, F., YAN, D.-M., LU, L., AND YANG, C. 2009. On centroidal Voronoi tessellation—energy smoothness and fast computation. *ACM Transactions on Graphics 28*, 4, 1–17.

LLOYD, S. P. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*, 2, 129–137.

MARÉCHAL, L. 2009. Advances in octree-based all-hexahedral mesh generation: handling sharp features. In *18th International Meshing Roundtable conference proceedings*.

MARTIN, T., COHEN, E., AND KIRBY, M. 2008. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. In *SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*, 269–280.

MESHKAT, S., AND TALMOR, D. 2000. Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh. *International Journal for Numerical Methods in Engineering 49*, 17–30.

MINKA, T. 1997. Old and new matrix algebra useful for statistics. Tech. rep., MIT Media Lab. revised 12/00.

MOUNT, D. M., AND ARYA, S. 1997. ANN: A library for approximate nearest neighbor searching. In *Proceedings CGC Workshop on Computational Geometry*, 33–40.

OWEN, S. J., AND SAIGAL, S. 2000. H-Morph: An indirect approach to advancing front hex meshing. *International Journal for Numerical Methods in Engineering 49*, 289–312.

RAY, N., LI, W. C., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Transactions on Graphics 25*, 4, 1460–1485.

SHEPHERD, J. F., AND JOHNSON, C. R. 2008. Hexahedral mesh generation constraints. *Engineering with Computers 24*, 3, 195–213.

SHEPHERD, J., MITCHELL, S. A., KNUPP, P., AND WHITE, D. 2000. Methods for multisweep automation. In *9th Intl. Meshing Roundtable conference proceedings*.

STATEN, M. L., OWEN, S. J., AND BLACKER, T. D. 2005. Unconstrained paving and plastering: A new idea for all hexahedral mesh generation. In *International Meshing Roundtable conference proceedings*.

STEINER, D., AND FISCHER, A. 2005. Planar parameterization for closed manifold genus-g meshes using any type of positive weights. *Journal of Computational Methods in Information Science and Engineering 5*, 2.

TCHON, K.-F., AND CAMARERO, R. 2006. Quad-dominant mesh adaptation using specialized simplicial optimization. In *15th International Meshing Roundtable conference proceedings*.

TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, 201–210.

TOURNOIS, J., WORMSER, C., ALLIEZ, P., AND DESBRUN, M. 2009. Interleaving delaunay refinement and optimization for practical isotropic tetrahedron mesh generation. *ACM Transactions on Graphics 28*, 75:1–75:9.

VALETTE, S., CHASSERY, J.-M., AND PROST, R. 2008. Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics 14*, 2, 369–381.

VYAS, V., AND SHIMADA, K. 2009. Tensor-guided hex-dominant mesh generation with targeted all-hex regions. In *18th International Meshing Roundtable conference proceedings*.

YAMAKAWA, S., AND SHIMADA, K. 2003. Anisotropic tetrahedral meshing via bubble packing and advancing front. *International Journal for Numerical Methods in Engineering 57*.

YAMAKAWA, S., AND SHIMADA, K. 2003. Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells. *International Journal for Numerical Methods in Engineering 57*, 2099–2129.

YAN, D.-M., LÉVY, B., LIU, Y., SUN, F., AND WANG, W. 2009. Isotropic remeshing with fast and exact computation of restricted Voronoi diagram. *Computer Graphics Forum 28*, 5, 1445–1454.